Development and Launch of a Hand-Gesture Controlled Mouse

[1] Annaluri Durgabhavani, [2] G.Hanumanthrao,

[1,] PG Student, Department of CSE, MAM Womens Engineering College, Kesanupalli, Narasaraopet, Palnadu District.

[2] Associate Professor, Department of CSE, MAM Womens Engineering College, Kesanupalli, Narasaraopet, Palnadu District.

Abstract:

Since computers first appeared on the scene, there has been a dramatic shift in how people and computers interact. The mouse is one of the most groundbreaking tools for HCI. Wireless and Bluetooth mice are examples of technological progress, but they still need other devices to function. Take, for instance, a Bluetooth mouse. It often comes with a charging cable and needs batteries. Some people find that these gadgets are less easy to operate since they need additional components. A virtual mouse technology that can solve these issues is introduced in this research. Computer vision and hand gestures are the backbone of the proposed system, which is based on HCI concepts. A webcam or integrated camera is used to capture gestures, which are then analyzed using methods such as color segmentation and detection. Players control the cursor's movements—left click, right click, double click—and scroll up and down using precise hand motions by donning colorful caps on their fingers. In order to monitor the user, the system takes video frames from the camera, analyzes them, and then uses the user's motions to control the mouse. This virtual mouse system provides a device-free alternative by eliminating the need for real devices, which helps progress HCI technology.

Keywords:

Color Detection, Hand Gesture Recognition, Human-Computer Interaction, and Hue Saturation Value.

1. Introduction

shrink to a more compact form. There are now wireless gadgets and dormant ones. This study lays forth a plan for the future of human-computer interaction (HCI) that might put certain gadgets into a dormant state. The proposed solution involves creating a virtual mouse that recognizes and responds to hand gestures. The goal is to eliminate the need for a conventional mouse by controlling the cursor functionalities using a basic camera. The Virtual Mouse bridges the gap between the computer and the user via the use of a single camera. It allows the user to operate the mouse's operations and interact with the system without using any physical or mechanical components. By using a webcam or in-built camera with a colored cap or colored sticky note paper, this gesture recognition system can easily catch and follow the fingertip of a hand, allowing the system to follow the hand's movement and color as it moves the cursor.

The majority of our interactions with computers take place via small peripherals like a mouse, keyboard, and the computer itself. In this study, the user's bare hand is the only input option utilizing a camera; nonetheless, the wireless devices do need a power supply and connection technology. This method of controlling the mouse pointer is, therefore, very interactive.

This system is built in Python using OpenCV, a package based on computer vision. The standard mouse and computer remote controls may soon be obsolete thanks to this solution. The only constraint is the state of the illumination. That's why the technology can't fully replace the conventional mouse just now. since most people use their PCs in dimly lit rooms.

A. Overview and Description of the Problem

The camera has to be placed so that it can observe the user's hands in the correct locations in order to detect fingers as a moving object and use it for mouse functionalities. For patients without limb control, in tight quarters, or in similar circumstances, this may be a lifesaver. This mouse is not actual, but rather virtual, and it follows your colorful fingers and camera frames. The Importance in Practical Settings (B) These days, a lot of people use video conferencing. Because of this, the vast majority of people who own computers also own laptops with built-in webcams. By using a camera, the suggested system has the potential to partly do away with the need of a mouse. As a kind of human-computer interaction (HCI), the use of hand gestures to control a computer is both novel and promising. This curiosity has been the subject of some excellent studies. Sign language recognition also makes use of hand gesture recognition technologies. C. Purpose An alternate method of controlling a mouse cursor has to be designed and put into action. Hand gesture recognition utilizing a camera and a color detection algorithm is an other approach. The end goal of this work is to create a system that can leverage any computer's color detection technology to identify hand gestures and control the mouse cursor.

2. Related Work on This Theory

Wearing a DataGlove is the norm for cursor control applications, however there are numerous other uses for hand gestures. Because of this, the user-system performance is diminished. Another problem with this procedure is the intricacy of the system. One approach to human-computer interaction (HCI) gesture detection might rely on hardware, while the other could use computer vision. Quam (1990) suggested a hardware-based solution that required the user to wear a cumbersome DataGlove [1]. Even while this approach provides highly precise control, it is not only difficult to use, but also impractical for most people in their daily lives since not everyone can utilize the same gestures. Both marker-based and non-marker based vision-based hand gesture recognition exist. In most cases, marker-based identification is much more accurate than non-marker based recognition. A basic colored cap on the fingertip is all that's needed for maker-based recognition, yet it outperforms competing gesture recognition systems in terms of accuracy. However, as compared to the DataGlove of the hardware-based system, this method is much lighter and hardly noticeable.

One possible future for computers to comprehend human gesture (Body Language) is gesture recognition. Rather of relying just on text, it will enable more complex interactions between humans and computers. When tracking gestures, the majority of marker-based gesture recognition mice employ two or more colored markers. The system becomes sluggish and displays certain delays during performance as a result of detecting numerous colors. "A Real-Time Hand Gesture Recognition System Using Motion History Image" [2] was suggested by ChenChiung Hsieh and Dung-Hua Liou in 2010. It relies on an adaptive skin color model and motion history image (MHI). They used a technique for detecting the direction of hand movements in motion history images and an adaptive skin color model in their study. The main drawback of the article is an issue in recognizing more complex hand movements. "A Human-Machine Interaction Technique: Hand Gesture Recognition Based on Hidden Markov Models with Trajectory of Hand Motion" [3], published in 2011 by Chang -Yi Kao and Chin-Shyurng Fahn, is essentially a learning-based method of human-machine interaction. Their work is spot on, however it was only compatible with powerful PCs. The authors of the 2013 article "Real Time Static and Dynamic Hand Gesture Recognition" [4] set out to create and research a real-world framework for gesture recognition that would find utility in a wide range of HCI applications. However, it could only be computed in well-lit environments and failed to function with complicated backgrounds. In their 2013 work "Cursor Control System Using Hand Gesture Recognition" [5], Ashwini M. Patil, Sneha U. Dudhane, and Monika B. Gandhi created a machine-user interface that uses basic computer vision and multimedia methods to recognize hand gestures. However, one big catch is that you have to recognize skin pixels and segment hands from recorded frames before you can work with gesture comparison algorithms. Methodology in the 2014 article "Mouse Control using a Web Camera based on Color Detection" [6] by Abhik Banerjee and Abhirup Ghosh is as follows: hand motions were captured using a camera based on color detection approach. Operating rooms must have a light backdrop free of brightly colored things; this is one of the constraints of their job. On some very powerful PCs, it performs well. By directly extracting fingers from prominent hand edges, Yimin Zhou, Guolai Jiang, and Yaorong Lin published "A novel finger and hand pose estimation technique for real-time hand gesture recognition" [7] in 2016. Segmenting and describing the hand posture based on the finger positions, palm center placement, and wrist position takes into account the geometrical properties of the hand. Unfortunately, this strategy is incompatible with low-end computers.

"Cursor Control using Hand Gestures" [8], created by Pooja Kumari, Saurabh Singh, and Vinay Kr. Pasi in 2016, relies on a system of several color bands, with each hue representing a distinct action. The key to performing mouse tasks is the quantity of colors. However, the system was controlled using a variety of colors. A function is delegated based on the quantity of colors rather than on individual motions. "Hand Gesture Recognition for Human-Computer Interaction" [9] was created in 2017 by Aashni Haria, Archanasri Subramanian, Nivedhitha Asokkumar, Shristi Poddar, and Jyothi S Nayak using a system that detects contours and extracts backgrounds. Working with it, however, is a real slog. An article titled "Virtual Mouse Using Hand Gesture" [10] was published in 2018 by Abhilash SS, Lisho Thomas, Naveen Wilson, and Chaithanya C. The study was created to operate with a color detecting system and is based on the amount of colors detected. However, it is limited to a small set of mouse motions and is inoperable in environments lacking a steady backdrop.

3. Methodology

Each component of the proposed system is detailed in its own section of this study.

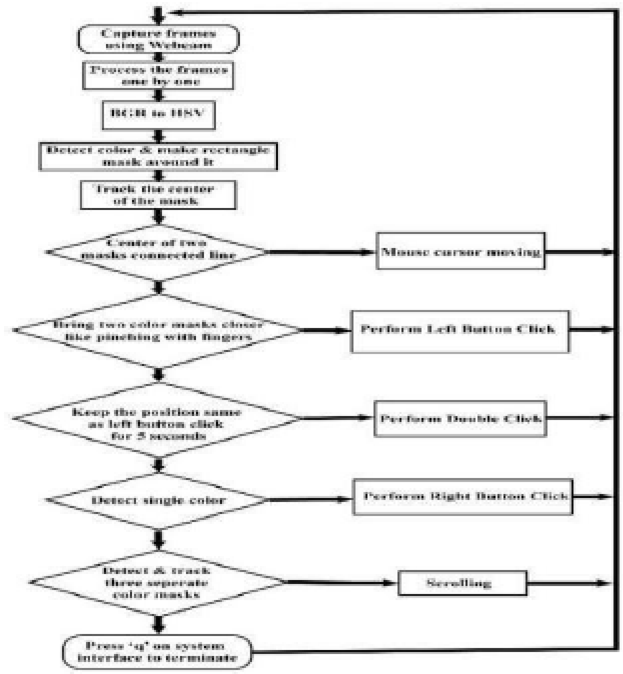


Fig. 1. Flowchart of The Methods of Gesture Based Mouse

A. Camera the system is designed to work with frames captured by a webcam or the inbuilt camera of a laptop. A video capture object is initialized so that the system can handle webcam input in real-time. By setting the device index to "0," we may use a single camera in this configuration. The device indices may be gradually adjusted to 1, 2, and so on to accommodate the installation of more cameras. The camera takes a series of pictures and sends them all to the computer for processing.

B. Snatching With the help of an endless loop, the camera records every single frame till the program ends. A transformation is performed on the video frames in real-time to convert them from BGR to HSV color space.

C. Identifying and Masking Colors The proposed method uses the webcam's collected frames to identify colors by identifying the colored pixels of fingers covered with color caps. The suggested system is built upon this stage. The technique yields a grayscale picture where the color cap is highlighted by pixel intensity, which differentiates it from the remainder of the frame. The next step in monitoring the color cap is to construct rectangular bounding boxes (masks) around it. When the location of these color caps is tracked upwards, the motion is recognized and the scrolling action is performed. The location and coordinates of the color caps are updated whenever three fingers are moved up or down. Scrolls are executed once all three color caps get updated coordinates. Scrolling down will be executed when the values of those y coordinates drop, and scrolling up will be executed when the values of those coordinates rise.

4. Gesture Recognition

A. Controlling the Mouse First, using the coordinates of the detected rectangle's center, we calculate the center of two colored objects. Using the following equation to get the midpoint and the built-in OpenCV function to draw a line between the two points:

$$M = \left(\frac{Xa+Xb}{2}, \frac{Ya+Yb}{2}\right) \quad (1)$$

To keep the mouse cursor on this midway, you may think of it as a tracker. This technology converts the coordinates from the resolution of the camera-captured frames to the resolution of the screen. The mouse cursor is pre-positioned to go to a certain area when it reaches that

at the open gesture posture, the mouse began to function. The user may control the movement of the mouse cursor in this way. (B) Clicking the Mouse To trigger click events, the suggested method employs near-touch motions. A new bounding box is generated using the tracking bounding boxes' edges if two rectangles' bounding boxes become too near to one another. At 20% of its initial size, the system triggers a left-button click, allowing the user to click on the newly-created bounding box. The user may conduct a double click by maintaining this position for more than 5 seconds. Additionally, the open motion is used to click the right button again. It just takes one finger to click the proper button. After identifying the color of one fingertip cap, the system will click the correct button. C. Scrolling the Mouse Using the open gesture movement with three fingers and color caps, the user may scroll through this system. The user may scroll down by bringing it to a downward position and using all three fingers at once. Similarly, it will scroll up if its position is adjusted to upwards. The location and coordinates of the color caps are updated whenever three fingers are moved up or down. Scrolls are executed once all three color caps get updated coordinates. Scrolling down will be executed when the values of those y coordinates drop, and scrolling up will be executed when the values of those coordinates rise.

5. Result & Evaluationv

We have used computer vision and Human Computer Interaction (HCI) in this study to make a contribution to future vision-based machine-human interaction. Using hand gestures to control mouse functionalities is the topic of the proposed article. Mouse movement, clicking the left and right buttons, double-clicking, and scrolling up and down are the primary controls. Users have a color palette from which to choose with this system. Depending on the backdrop and illumination, users may choose a color from a small set of predetermined bands. In other contexts, this might look different. At startup, for instance, the user will see a color palette that includes green, yellow, red, blue, and two more options. Either the user must choose a color that stands out against the backdrop or select a color that blends in with the existing backdrop.

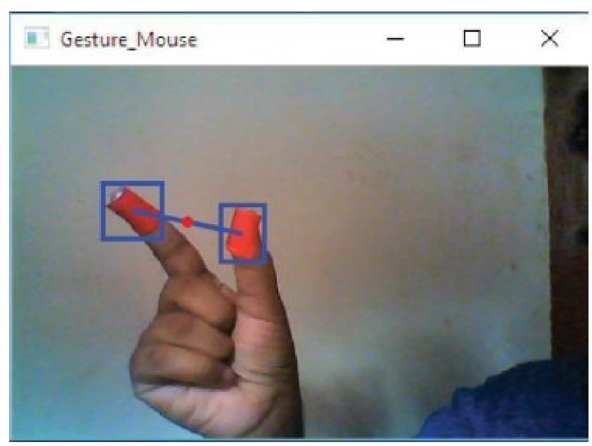


Fig. 2. Mouse Movement (Open Gesture)

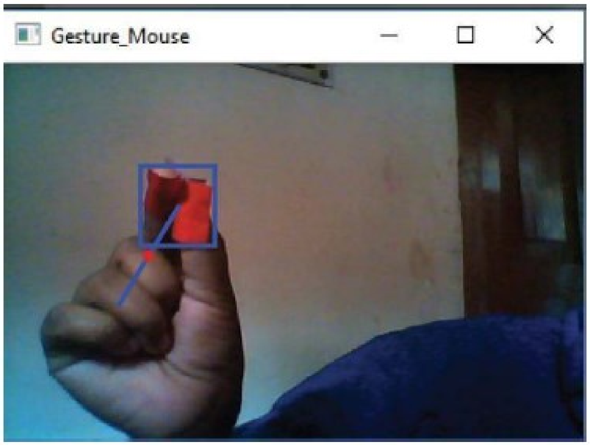


Fig. 3. Left Button Click (Close Gesture)

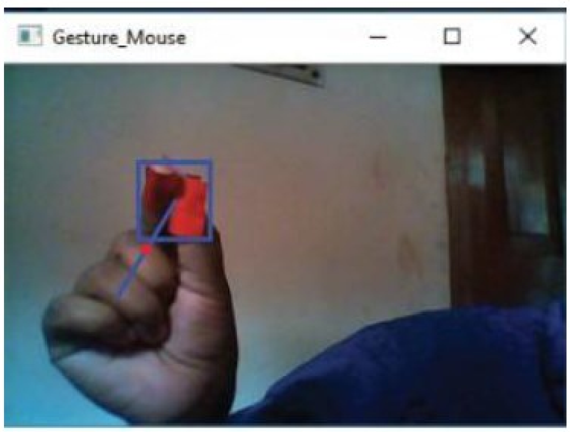


Fig. 4. 5 Seconds Perform Double Click (Close Gesture)

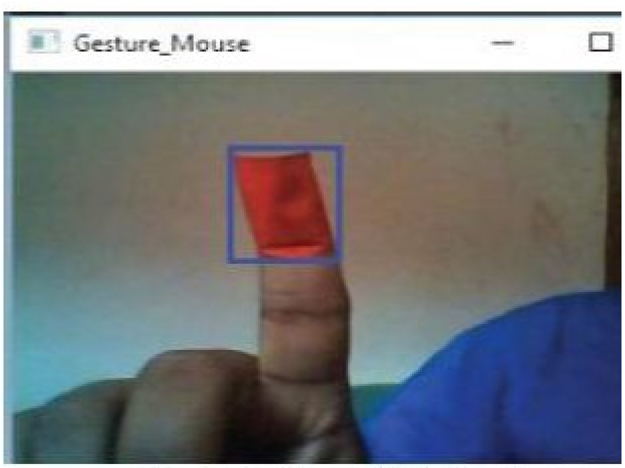


Fig. 5. Right Button Click (Open Gesture)

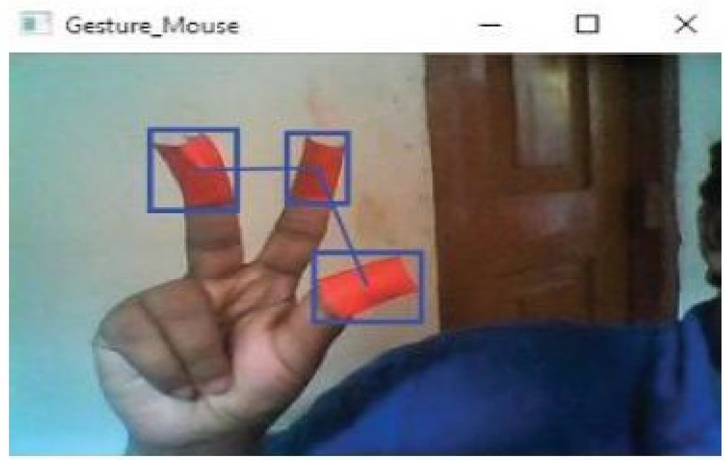


Fig. 6. Scrolling Up & Down (Open Gesture)

All of these tests are conducted on a white backdrop. With a speed of 78 to 90%, it gets the job done quickly. Additionally, it has been tested with a variety of complicated backdrops, including checkered shirts, colorful t-shirts, complex home backgrounds, indoor daylight surroundings, and light environments inside the house, among others. It's worth noting that the majority of individuals who sit in front of computers wear plain shirts or t-shirts. An accurate evaluation of these occurrences is shown in a table.

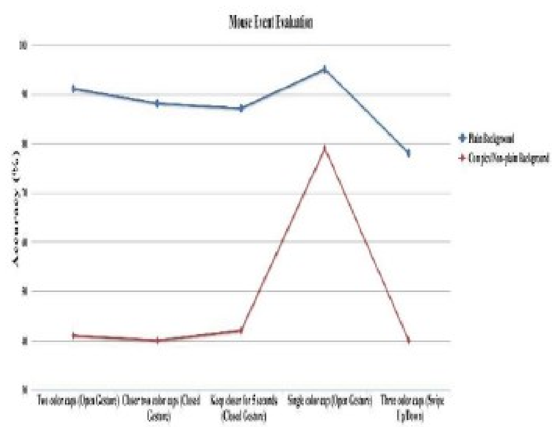
TABLE I. MOUSE EVENTS & EVALUATIONS

Gesture Input	Evaluation		
	Mouse events	*Accuracy with plain background (in %)*	*Accuracy with complex/ non-plain background (in %)*
Two color caps (Open Gesture)	Mouse Movement	91	41
Closer two color caps (Closed Gesture)	Left Button Click	88	40
Keep closer for 5 seconds (Closed Gesture)	Double Click	87	42
Single color cap (Open Gesture)	Right Button click	95	79
Three color caps (Swipe Up/Down)	Scrolling up or down	78	40

On such a complicated and uneven backdrop, the system's inefficiency becomes immediately apparent. However, as compared to other systems that employ gestures to operate the mouse, ours is noticeably quicker. By using a high resolution camera, this precision may be enhanced. The data in the table are derived from the Video Graphics Array (VGA) camera that is integrated inside the laptop. A Windows PC may not always be the best platform to run this system on; in some cases, a Linux or Macintosh machine may provide superior results.

TABLE II. COMPARISON WITH EXISTING SYSTEMS

Existing methods	Comparison		
	Recognized Gestures	*Average Accuracy (in %)*	*Control Type (Static/Dynamic)*
Color tracking & counting[8]	Five (05) : Mouse Movement, Left Click, Right Click & Double Click etc.	Not Available	Dynamic
Finger & palm tracking[9]	Seven (07) : Opening Media Player, Web Page, Launching Powerpoint etc.	78	Static
Color masking & pinchflag [10]	Three (03) : Mouse Movement, Left Button Click, File Transfer	Not Available	Dynamic
Our proposed method	Five (05) : Mouse Movement, Left Click, Right Click, Double Click, Scrolling	78	Dynamic



6. Application

This project is great for presentations and helps reduce the amount of space needed in the office and the number of pieces of hardware that are needed. The technology proves reliable in high-pressure environments, providing a smoother experience compared to other contemporary computer interface methods. The elimination of peripheral gadgets allows for a more intimate and productive relationship between the person and their work area. Important uses: • This system has many important uses, one of which being robot controlling. A great enhancement to this technology might be the ability to control robots without the need for additional machinery or gadgets. Digital artists may utilize this gesture-based mouse on digital canvases to sketch 2D or 3D creations. As a result, artists will have more leeway, energy, and room to work. Using a motion mouse, you may manage vital events such as a battlefield, operating room, or mining field. • You can play VR/AR games more easily with only your bare hands, without the need for any additional gadgets or cords. • People who are unable to control their limbs may find this method to be quite helpful. Individuals who are deaf or hard of hearing may find this mouse method useful for sign language. It may facilitate their interaction with computers.

7. Conclusion

One such technology is the virtual gesture control mouse, which uses a webcam to control the pointer and other mouse operations in real time. Included in our solution are the standard mouse actions—left click, right click, double click, and scrolling—as well as the ability to pick icons. To make navigating icons and moving the pointer easier, the system employs picture comparison and motion detection technologies. The results prove that the algorithms work well with sufficient illumination and a high-quality camera. We want to improve the system in the future by adding capabilities like multi-window interaction, window resizing, and multi-finger motions for shutting windows.

References

1. D.L. Quam, “Gesture recognition with a Data Glove,” 10.1109/NAECON.1990.112862, vol. 2, pp. 755 - 760, 1990.

2. C.-Chiung Hsieh, D.-Hua Liou, & D. Lee, , “A real time hand gesture recognition system using motion history image,” Proc.

IEEE Int'l Conf. Signal Processing Systems (ICSPS),

3. 10.1109/ICSPS.2010.5555462, 2010.

4. C.-Y. Kao, C.-S. Fahn, "A Human-Machine Interaction Technique: Hand Gesture Recognition Based on Hidden Markov Models with Trajectory of Hand Motion," in Procedia Engineering 15 pp. 3739 – 3743, 2011.

5. P.S. Neethu,"Real Time Static & Dynamic Hand Gesture Recognition," International Journal of Scientific & Engineering Research, Vol. 4, Issue3, March 2013.

   A. M. Patil1, S. U. Dudhane1, M. B. Gandhi1," Cursor Control System Using Hand Gesture Recognition", International journal of advanced research in computer and communication engineering, vol. 2, issue: 5, May 2013.

6. Banerjee, A. Ghosh, K. Bharadwaj, "Mouse Control using a Web Camera based on Color Detection,"IJCTT,vol.9, March 2014.

7. Y. Zhou, G. Jiang & Y. Lin, "A novel finger and hand pose estimation technique for real-time hand gesture recognition," in Pattern Recognition vol. 49, pp. 102-114, January 2016.

8. P. Kumari, S. Singh & V. Kr. Pasi, "Cursor Control using Hand Gestures," Recent Trends in Future Prospective in Engineering & Management Technology 2016, International Journal of Computer Applications (0975 – 8887), 2016.

   A. Haria, A. Subramanian, N. Asokkumar, S. Poddar & J. S. Nayak, "Hand Gesture Recognition for Human

9. Advances in Computing & Communications, ICACC-2017, Cochin, India, Procedia Computer Science 115 (2017), pp.367–374, 22-24, August 2017.

10. Singh, Harsh Pratap, et al. "AVATRY: Virtual Fitting Room Solution." 2024 2nd International Conference on Computer, Communication and Control (IC4). IEEE, 2024.

11. Singh, Nagendra, et al. "Blockchain Cloud Computing: Comparative study on DDoS, MITM and SQL Injection Attack." 2024 IEEE International Conference on Big Data & Machine Learning (ICBDML). IEEE, 2024.

12. Singh, Harsh Pratap, et al. "Logistic Regression based Sentiment Analysis System: Rectify." 2024 IEEE International Conference on Big Data & Machine Learning (ICBDML). IEEE, 2024.

13. Naiyer, Vaseem, Jitendra Sheetlani, and Harsh Pratap Singh. "Software Quality Prediction Using Machine Learning Application." Smart Intelligent Computing and Applications: Proceedings of the Third International Conference on Smart Computing and Informatics, Volume 2. Springer Singapore, 2020.

14. M.joshi, S.Birla, H.Pal, K. Khatri, M.Kadwal, D. Salitra, "AI-Driven Intrusion Detection System: Leveraging Deep Learning for Network Security" Nanotechnology Perceptions ISSN 1660-6795 Q4 Journal with SJR 0.11 Volume Vol.20, S10 Sept 2024.

15. K. Paul Joshua, D. Srinivasa Rao, Govinda Patil, Mohit Kadwal, Jitendra Choudhary, "Cascaded ANN Based Clustering and Optimized Routing Path Selection in Mobile Adhoc Networks" SSRG International Journal of Electrical and Electronics Engineering Volume 10 Issue 6, 81-93, June 2023.